PAGE RANK: HOW DOES GOOGLE WORK? 15

Alright guys in the previous videos we had seen what is a directed graph. how can we work with networkx and directed graph and what is points distribution methods and how do we, we have simulated it with excel sheet and there was for a small network and now we had started off with the programming part complex network having lots of nodes and edges. We had created a random network we had visualised it we had initially given equal points to all the nodes and we had started the game and we had asked the nodes to distribute the points based on their outgoing edges based on structure they keep distributing and we decide when to stop this game. Just because in excel we have that flexibility we can drag the calculations as long as we want so something like that same like that i want to give the control to the user that's why we had done it something like that but the user will decide when to stop, whenever the user stops just it will just say it is the final values something we had got it right so now what should we do once you get the final values you should wrap the nodes based on those values and next step would be compare it with the actual page rank values that is provided by the built in functionality in netwotkx we will do that so after they are done distributing we got the final points so i have to what should i do rank by points right i have to rank the nodes by points so let me say rank by points let me call it so to rank the nodes by points what do i need? I want to final points so i want this so i have to rank the nodes by points. Ok as you could see it undefined name see it is the warning message that is given so later define it let u define it ok define this thing and let me see how shall we start with it so we have decide this functionality rank by points so first thing is if you just like that we have points is nothing but list right so if we sort the list its simple and you just rank it that's what you are saying but if you sort the list you will lose the information as to this point belongs to which node so here we need to keep track of two information which node has got how many points it's not just about how many points it's about which node has got how many points so we have to convert this list structure into a dictionary structure we have to convert the list structure into a dictionary structure and this dictionary structure will be applying the sort functionality that is defined on dictionaries and this will be using to display the rank of the nodes based o n the points they have accumulated alright so as i said i need a dictionary let me say d so i have to have the node the points are actually arranged in the same order know zero point is the zeroth element of this list node ones point is the first element of the list something like that, that's how it has been arrange so let me it's just enough for loop one for a loop over i in range length of points so whatever is the index that is the node ideas as well in our case length of final points basically final points or may be even we can call it as a points as well so there it was final points and even here you can call it as points so for i in range length of points that is i am going to iterate over the list i am going to convert it into a dictionary, dictionary with key i the list index is what i am going to make it as key so that is same as node id right so i am going to make it as a key is nothing the corresponding value of points at that index. So i have created the dictionary now i should sort this dictionary to sort the dictionary we have built in functionality sorted you need to pass the items of the dictionary and here dictionary has too things keys and value we need to sorted based on the value part right so for that you need to say key equal to lambda so key here is nothing but based on what value you need to do, what is the factor based on which you need to sort. That is what you call as key and

lambda is built in functionality in python basically it like for example alphabetical order ascending order attendance order or whatever you can call it as how would you do it? Based on the first letter you will sort the names A comes on the beginning then comes B then comes C so on. What if you have different demand? You want to sort it based on the last letter, so in that case you have to make use of custom function that you are actually going to write based on that function you have to sort on your requirement something like that if you have some other factor that you are defining and based on that factor you want sorting to happen you will use this kind of a syntax. And lambda because i am not going to define a new requirement that is not present in python this sort by value is a common requirement so python has defined it so it will say the upper f let me call it f nothing but at f at index one so key is at index zero and value is at index one so this lambda functionality sort of converts it into a tupple format and it will take the value at index one and it will sort based on it so this is the defined functionality that has been defined in python itself in case you want it an new requirement here you can say key equal to my function and based on that function it will sort so what is the factor i should considered for sorting that is what is denoted by this key ok so my factor is nothing but i wanted to be based on the first index value that is based on the value of points i want the sorting to take place the relationship between the node id and the points is also preferred and sorting also occurs so this is the specialty so if once it is sorted it will return a tupple of list of tupples basically so i need to print it so i will print this value so even that i have sorted and printed so the ranking part is done. So what should i do now? Once i have ranked let me compare it with the default networkx function i am going to use that and i am going to compare that results. So the default function of networkx is nx dot page rank and you have to pass the graph and it will return a dictionary basically. So that's why we had also used a dictionary right we had why dictionary? Because you need reserve you have to actually preserve the fact that this node has this many points you have to preserve that fact. So let u say this is the result of your page rank result and dictionary nx dot page rank do G just you need to give the value of page rank these things maximum number of iterations how many iterations you want you can customise it you can do it as per your demands but even you don't give there are some default values they would be taken and it would be calculated so we got the resultant that the dictionary and same like this i have to sort and print the sorted values. Right? So let me copy paste this lines so let me just copy and paste it so i will copy given that i have copied let me paste the line here and instead of B it is the name of my dictionary is result so i have taken a dictionary and based on the values i want the sorting to occur that is based on the points accumulated i want the sorting to occur so just a quick recap and then we proceed with running. So we had seen till distribution of points the output even is displayed here so before proceeding further i will clear the screen and i will run so the output is as well displayed here it gets converged to some specific distribution we have seen at till there and once we have got the convergence state we have to rank the node this is the points for the node zero this is for node one, node two and so on so we have to rank the nodes based on it just keep this is the list if you just apply sort on list the fact that this node had this points is lost that's why we will convert it into dictionary as you could see here we are converting the list into dictionary preserving the fact that this node has this many points and we are applying this built in functionality sorted on dictionary and this is the functionality that take care of sorting based on a given parameter based on the pints

value we want to sort it so we are using that built in functionality just that to preserve the part of his node has these many points you are using the this many data structure you are converting the list into dictionary. Also we have done this for our method points distribution method and once it is done we are going to compare it with the default networkx functionality page rank that will also return the dictionary and we are applying the same sorting mechanism and we are displaying the sorted result we will see how is the result we get after the points distribution method and what is that built in result that is obtained will see how it is same or different or what is the thing that is the node that is ranked as number one here it is the same built in as well let us see all those things before running let me clear the screen and let me run for you let me run now so as i have got the some network some distribution will continue continue continue continue continue i keep continuing i keep continuing yes i guess i got a convergence see because i was saying these two values are failure there is convergence there is convergence here let me press hash to stop ok i have got convergence here ok so this is the value we have got based on the points distribution method node seven is the least rank node and node six is the highest rank node node seven and node six yeah same we are getting we are getting different values here because as i have said take a different seed value the final value will be different but as i have said if you take the this particular value by sum of all these values if you find the distribution will be same that is here if you take this particular value twelve point something by three hundred if you take the particular fraction if you take the fraction that will be same for both the things you can verify it this built in functionality is using some other methodology also i hope you have seen or you will be seeing another method of random walk to get the page rank values so it is using random walk and some other methodologies and some other seed values it is built in functionality is doing so the values may differ but the ranking if you see node seven is the least rank node node six is the highest rank node that i preserved not only that the ordering is exactly preserved as you could see, seven four zero eight seven four zero eight ok two nine one three five six two nine one three five six wow this is exactly matching so good it is but our method of point distribution the built in page rank method exactly matches as i say whatever be the network structure there is a convergence state there is a convergence configuration this is mathematical proof that you can take it as an advance material from this lecture. You can look it up some amount of probability is the pre requisite for the mathematical part you can look it up as an advance material from the its very required if you are interested and if you are key person and math oriented person highly enthuastic with maths you can look it up requisites are probability theory that’s it you can look it up and see to it, it will always converge there is a vigorous mathematical proof for it. Alright guys i hope you enjoyed this. Activity on page rank using point’s distribution method, thanks for watching have a nice day.